Toon Shader Primer

Michael Arias, creator of the Toon Shaders, answers frequently asked questions about Toon rendering with SOFTIMAGE³|3D and mental ray

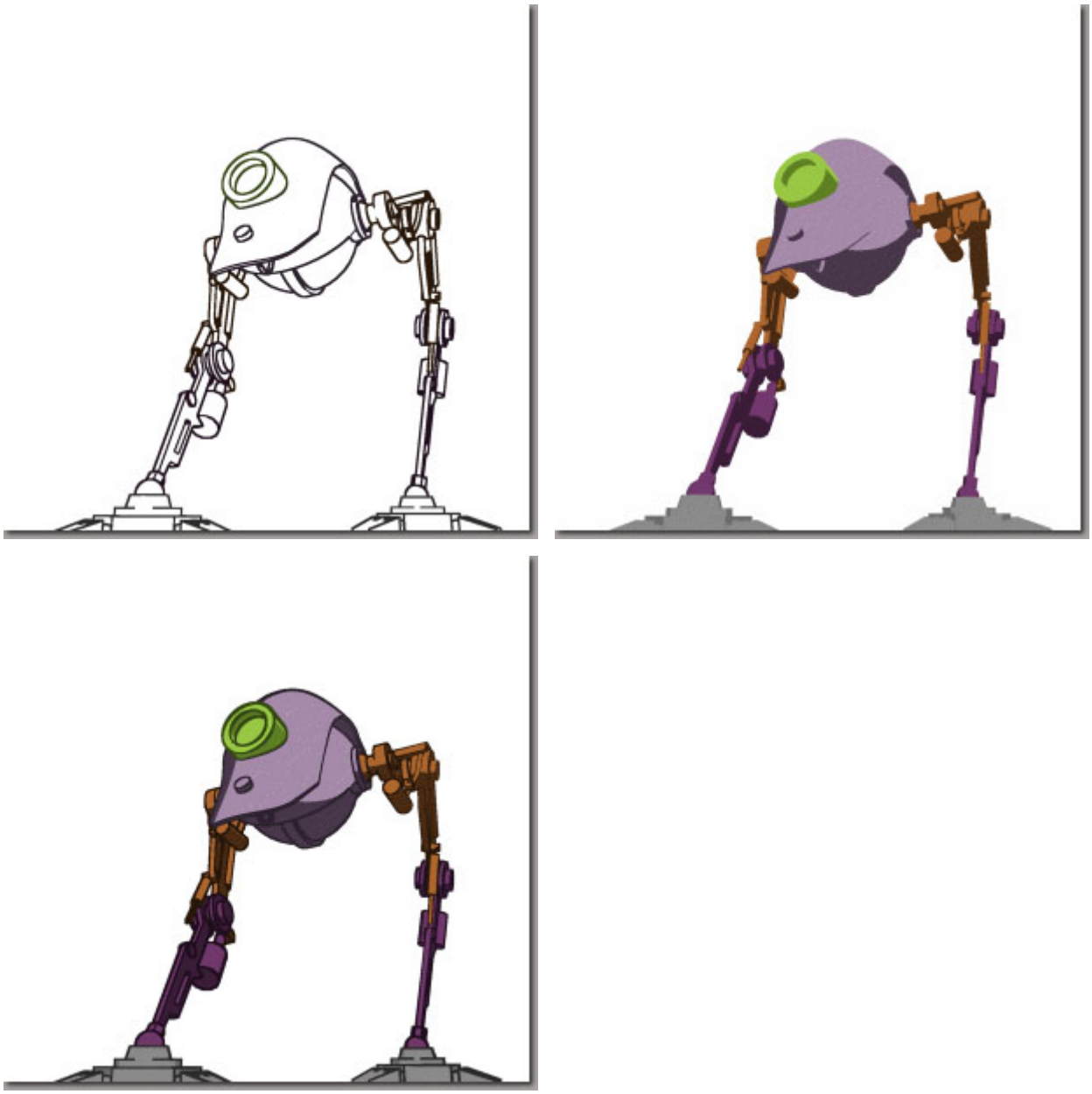
Index of Topics

1. What is Toon rendering?
2. What are Toon Shaders?
3. What is the ToonAssistant?
4. If the ToonAssistant does all of this, why would I apply Toon material shaders manually when using the ToonAssistant?
5. What antialiasing settings should I use when Toon rendering?
6. Why is Toon rendering slower than normal rendering?
7. Are there any special modeling techniques I should use for Toon rendering?
8. Can I use mental ray's motion blur when Toon rendering?
9. Can I use mental ray's depth of field when Toon rendering?
10. How do I go about setting up materials for Toon rendering?
11. Where will ink lines show up?
12. What color(s) will be used for the ink lines?
13. I have an object composed of several models, but I want it to appear as one model, without the "seams" visible where models intersect...
14. Why does that annoying "...existing material shaders will not be replaced" dialog appear every time I render with the ToonAssistant?
15. How should I go about simulating a "dark" Toon scene (something at night for example)?
16. What are the Ink Variation controls for?

---

1. What is Toon rendering?

"Toon rendering" with SOFTIMAGE|3D is a method of simulating the "look" of cel animation (i.e., "2D" or "traditional" animation) using 3D-animation techniques - common characteristics of cel animation, such as contour lines, outlines, and solid-color shading are all replicated. 3D models and animation may be rendered so that the results are more-or-less indistinguishable from images created using traditional cel techniques. Since the images we are imitating are produced using so-called "ink-and-paint" techniques, we refer to many of the 3D equivalents using similar terms: ink (instead of "contour lines), paint (shading), etc.



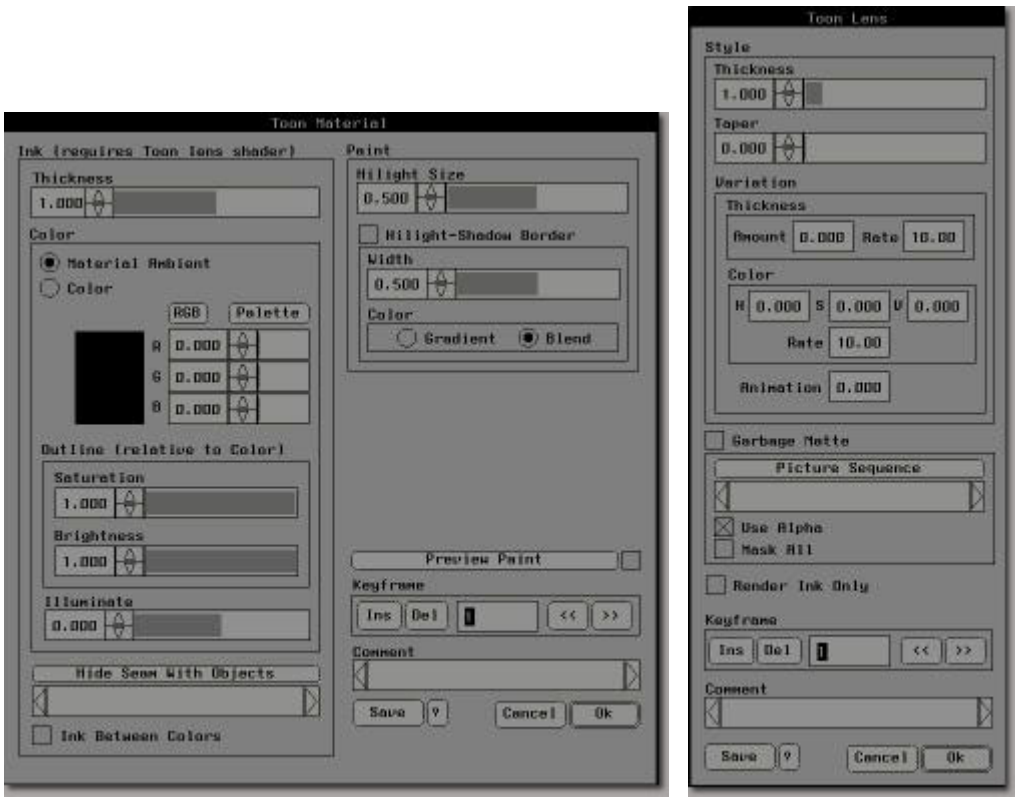
Toon ink, paint, and ink+paint

2. What are Toon Shaders?

To create a Toon rendering we use special programs called shaders. mental ray (the renderer), uses these to achieve the ink and paint effects. Shaders, like plugins for other software, enhance the basic capabilities of mental ray: When Toon rendering, instead of shading our models with Phong, Flat, or Lambert shading (the defaults), for example, we can now use "Toon shading" to create the flat-color hilights and shadows on surfaces.

There are two Toon shaders that we rely on most, a lens shader (for ink effects) and a material shader (for paint effects). These shaders work together to create a completed Toon rendering. For the most part one can think of the lens shader as a global control over ink: it allows control over line width, waviness, color, etc. The material shader, because it is applied to each individual model (or parts of a model if a polygonal mesh is being used), allows local control of Toon rendering attributes. That is, local control of both of the paint effect (size of hilight, style of hilight, etc.), and certain parameters of the ink used on each model (for example, each Toon material shader may specify a different line width, relative to the line width specified to the lens shader). Though the material shader may be used alone, without the lens shader, none of the

ink effects will be rendered without the lens shader. None of the lens shader's ink effects will render without the material shader applied to models, so the two shaders are usually are used together.



Toon material and lens shader interfaces

3. What is the ToonAssistant?

The ToonAssistant is a plugin for SOFTIMAGE|3D which offers a very convenient interface for "managing" the application of the shaders, allows previewing of the scene, and animation of the various rendering parameters. Using the ToonAssistant allows you to apply materials and textures as you normally would when using SOFTIMAGE|3D, then apply the Toon effects as a final step in rendering, without ever attaching lens or material shaders. When using the ToonAssistant it is not necessary to go through the tedious steps of applying shaders to each material, lens shaders to the camera, etc. When using the ToonAssistant, it is never necessary to apply the Toon lens shader, though there are some times when it may be useful to "manually" apply Toon material shaders to particular models. Like other plugins for SOFTIMAGE|3D, the ToonAssistant has a custom icon (a happy face). Once the ToonAssistant has been loaded, its settings may be adjusted by selecting the custom icon and choosing **Info>Custom Effect**.

ToonAssistant interface

ToonAssistant custom icon

4. If the ToonAssistant does all of this, why would I apply Toon material shaders manually when using the ToonAssistant?

As mentioned before, there are certain properties local to each material which may be manipulated from the Toon material shader interface, such as object grouping, line width, etc. If you get results you like using the ToonAssistant by itself, by all means don't bother with any Toon Shaders. If however, things look generally ok using the ToonAssistant, but you need to, for example, tweak the ink-line width of one model only, then you might apply the Toon material shader to that one model, and adjust the line width from the Toon material shader interface. When rendering with the ToonAssistant all existing Toon material shaders are left untouched, and their values used to alter the rendering properties for the models to which they are applied.

5. What antialiasing settings should I use when

Toon rendering?

This depends on the results you want for the rendered images. Higher levels of antialiasing produce better results. Insufficient antialiasing is particularly noticeable where ink lines are drawn: the lines chatter during animation (in some cases, the lines show the dreaded "jaggies"). If every other frame of the animation is rendered, then less antialiasing is required. The recommended settings are:

- Use mental ray's adaptive sampling algorithm. This minimizes wasted samples (on large empty areas of the frame).
- For medium-quality work with decent lines (sufficient for broadcast), set Min Samples to -1 and Max Samples to 2. When higher quality is required (film applications), set Min Samples to 0 and Max Samples to 2. For print work add the Gaussian filter with a width and height of 2 (this adds additional smoothness by blurring neighboring samples together, though rendering is much longer).
- Set low oversampling thresholds: perhaps 0.02, 0.02, 0.02.

6. Why is Toon rendering slower than normal rendering?

The Toon lens shader does a fair bit of sampling (i.e., firing of rays) above and beyond mental ray's normal eye-ray casting. On average Toon rendering requires about 5 times the usual number of rays be fired from the camera to render a frame. This is offset a bit by the fact that the shading computations for Toon rendering are much simpler than those required by Phong shading, for example. Also, since our ink is rendered during the sampling process, there is no post-processing time to take into account. If multiple processors/machines are being used co-operatively, then the processing time for ink lines will be divided among them (this would most likely not be the case if post processing were employed to render contours). Nonetheless, expect longer rendering times. This is an instance of another CG production proverb: "you get nothing for free" (this applies to life in capitalist nations as well).

7. Are there any special modeling techniques I should use for Toon rendering?

Few in particular. Most models can be Toon rendered with fairly good results. Most modelers have a preference for free-form surfaces over polygonal meshes, or vice-versa. Obviously, both have pros and cons. I find free-form surfaces (i.e., standard and NURBS patches) particularly good for Toon rendering because (among other things) they typically result in very smooth surfaces. This is important in simulating cel animation because most cel-animated characters and objects are drawn to look as though composed of fairly smooth surfaces.

To insure perfectly smooth surfaces when using patches you may specify that a surface be tessellated using a view-dependant method (i.e., you specify how many pixels wide each triangle resulting from patch subdivision will appear). To set the subdivision method for any model use **Info>Selection**, for groups of models use **Info>mental ray**. Polygonal surfaces, on the other hand, will show irregular contours at the edge of hilights if care is not used to create the illusion of smoothness.

When surfaces are smooth shaded, such as with Phong and Lambert shading, discontinuities in surface curvature are mostly hidden by the shading itself, though there are limits here as well: if very large polygons are used their flatness will become obvious. Unless a stylized "cutout" look is desired, polygon models look best when modeled with fairly high detail.

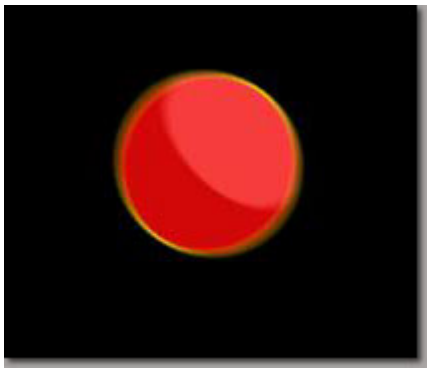
When polygon modeling for Toon rendering I usually set **Display** to show edge flags, and

sometimes even surface normals. It is also sometimes necessary to raise the automatic discontinuity threshold higher than the default 60 degrees. Where edge flags or discontinuities are shown in the SOFTIMAGE|3D modeling interface, ink lines will be drawn as well (in fact this can be used to advantage because it allows "modeling" of ink lines by toggling edge flags between adjacent polygons). All of the Viewpoint polygonal models that come as "gifts" with SOFTIMAGE|3D Toon render pretty nicely.

Low-polygon model

8. Can I use mental ray's motion blur when Toon rendering?

Certainly, both ink and paint effects will motion blur, but expect much longer rendering times than without motion blur (this is the case when not using Toon shaders as well).



Toon motion blur

9. Can I use mental ray's depth of field when Toon rendering?

Yes, both ink and paint effects can be rendered "out of focus" but, as ever, expect longer rendering times and be prepared to increase the level of antialiasing.

10. How do I go about setting up materials for Toon rendering?

The simplest method is to apply materials as one normally would when setting up a scene to render with SOFTIMAGE|3D (the same rules apply: Materials are inherited by children in a hierarchy, models with no material applied are rendered using a default material, etc.). Then, without assigning any Toon shaders per se, use the ToonAssistant for the rest.

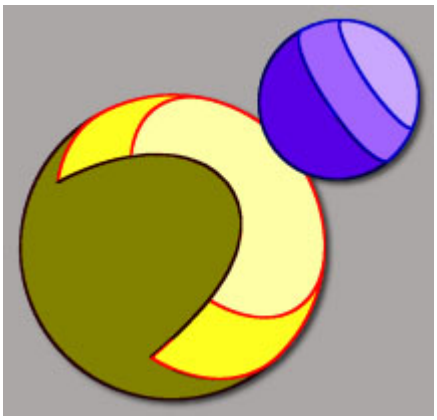
11. Where will ink lines show up?

The Toon lens shader "inks" the rendering in a few places:

- The outlines of objects (i.e., their silhouettes).
- The intersection of different models (this may be altered through the Toon material shader interface).
- Discontinuities in models' surfaces (i.e., areas of high curvature or depth from the camera, edge flags and automatic discontinuity on polygonal models).
- Between colors (an option only available through the Toon material shader interface).

12. What color(s) will be used for the ink lines?

Either with the ToonAssistant or the Toon material shader, two basic options are available: The ambient color assigned in the Material Editor may be used for the ink lines or a "custom" color. If the ToonAssistant is being used it is best to set each material's ambient color to one appropriate for the ink lines on the model (or collection of polys) to which the material is applied. Then, in the ToonAssistant's Ink Color tab, make sure Material Ambient is chosen for the Ink color. This allows each model to have its own ink color, as set in the Material Editor. In certain cases it may be desirable to use the custom ink-color swatch/sliders in the Toon material shader interface (or, alternately, ToonAssistant Ink Color tab) - if, for example, the material's ambient color is required to render shadows falling on a surface. In this case, unless the same ink color is to be used for all models, the Toon material shader interface should be used. In either case, in addition to the basic color set for ink lines, there are also options in both ToonAssistant and Toon material shader interfaces for drawing the outlines in a brighter/darker and less/more-saturated color than that used for internal contours. There are also options for modifying the color of ink lines based on whether the ink line being drawn is in or out of shadow.



Illuminated ink (shadowed ink is darker)

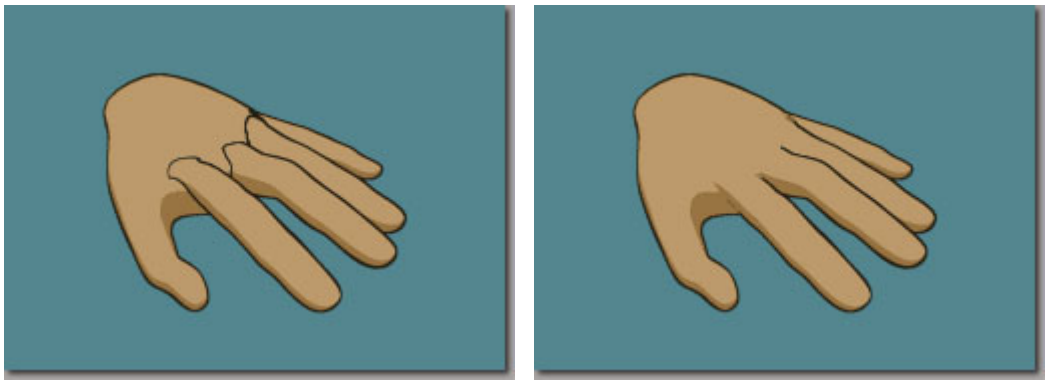
Ink Color tab

13. I have an object composed of several models, but I want it to appear as one model, without the "seams" visible where models intersect...

It is possible to group any number of models such that ink lines will not be drawn at their intersections with each other. This is an option available through the Toon material shader interface but not through the ToonAssistant interface (though it is of course compatible with the ToonAssistant tool). In the Ink section of the Toon material shader interface is a button with the title Hide Seam With Objects and below this, a text field. Clicking on the button calls up an browser which allows selection of one or more models (selecting more than one model is possible by holding down the Shift key while clicking on the various models' names).

In the following example we have a model of a hand made up of a separate model for each finger as well as for the thumb and palm. Normally seams between fingers and palm would be inked, but this is most likely not desirable. So to eliminate the objectionable seams it is necessary to attach the Toon material shader to the palm model and, using the model browser mentioned above, put the names of the finger and palm models in its Hide Seam With Objects field. Since we are only trying to eliminate the seams between palm and fingers (and not between each finger and its neighbors) we need only put the name of the palm model in each of the fingers' Hide Seam fields.

Actually, in this example we take advantage of the fingers being children of the palm model (thus inheriting its material automatically). Note that, though the ink is not drawn between the seams, discontinuities in surface curvature at the intersection of models will be visible as discontinuous illumination. This is remedied only by better modelling.



Palm model with and without seams

14. Why does that annoying "...existing material shaders will not be replaced" dialog appear every time I render with the ToonAssistant?

This informs you that some of the materials in the scene already have material shaders attached, and that the ToonAssistant will not replace these shaders. This is necessary to allow use of the Toon material shaders in the same scene as the ToonAssistant (for fine control, for example).

ToonAssistant warning dialog

15. How should I go about simulating a "dark" Toon scene (something at night for example)?

Usually two lighting qualities distinguish nighttime images from those simulating daylight: so-called "high-key lighting" and low lighting levels. The term "high key" refers to a high ratio of key light to fill (ambient) light, usually accomplished by having very dark shadows (in the CG world, low "ambient" illumination). "Low lighting" means what it sounds like, i.e., lights that produce little illumination, i.e., have fairly dark colors. There are a couple of things to keep in mind when replicating this in the Toon universe. First, hilights add, that is to say that where two lights create overlapping hilights on a surface the colors combine additively. It is common to have only one light source in a Toon scene, more so for night scenes where, for example, the moon might be the only light source. This is important to consider when creating a dark scene because multiple light sources will create brighter and brighter areas on surfaces that have overlapping hilights.

The hilight color on a surface is actually a product of both the color of illumination reaching the surface and the specular color set in the Material Editor dialog. Thus a red light (color 1, 0, 0)

hitting a surface with a material with a blue (0, 0, 1) specular color will have no visible hilights because the product of the illumination and specular colors is black! Another thing to consider is that dark scenes should have a very dark shadow color (the diffuse color in the Material Editor dialog). A surface with the Toon material shader will be colored with the shadow color when it is unlit. This is the darkest that the surface will be rendered.

"daylight" and "nighttime" lighting

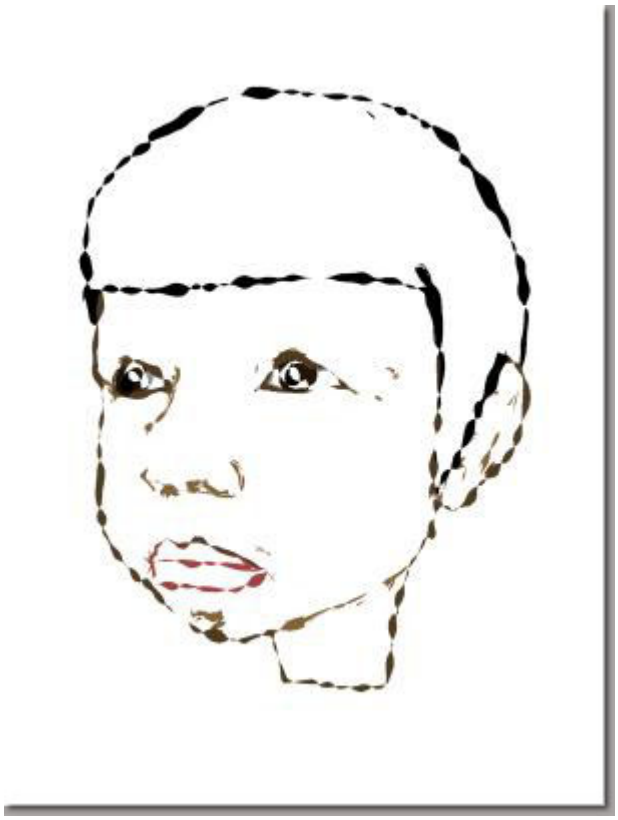
16. What are the Ink Variation controls for?

The parameters in the Ink Variation tab of the ToonAssistant or in the Toon lens shader allow you to randomly vary ink lines' thickness and color. This is useful for achieving a "human touch" to the renderings, as well as for simulating various common drawing types (i.e., oil paint, crayon, Japanese brush, etc.). In the case of thickness, one has control over the amount and frequency (rate) of the variation of the line thickness. The greater the amount, the greater the difference in thickness between a line at its thickest and thinnest states. A higher rate means a line with more "wiggles", while a lower frequency gives a more smoothly varying line.

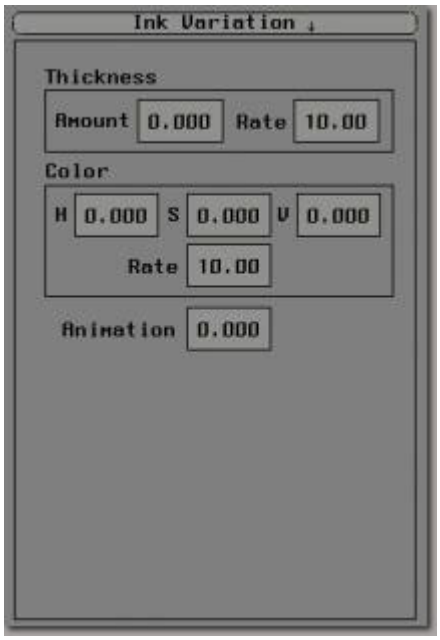
The same holds for color: different amounts of variation may be set for each of H (hue), L ("luminance" or brightness), and S (saturation), and a rate chosen for all three together. Lastly all of these random variations may be animated, either by keyframing the values, or by choosing an animation speed (the animation field) other than zero. A higher value for animation gives faster variations over time - a very high number will give a "chattering" line. One thing to keep in mind is that the effects of variation settings differ depending on the rendering resolution.

Japanese brush: varying Thickness (Amount = 6.0, Rate = 20.0)

Crayon: varying Thickness (Amount = 4.0, Rate = 1000.0)



Blobby oil paint: varying Thickness (Amount = 10.0, Rate = 50.0)



ToonAssistant Ink Variation tab

Toon Shader Primer was written by Michael Arias, Project Leader/Asia & Toon software development, SOFTIMAGE Special Projects.